STIG Manager 2

Reference Document

By Eric Hansen

Table of Contents

[About: 3](#_Toc35848164)

[Basic Steps: 3](#_Toc35848165)

[Search: 11](#_Toc35848166)

[STIGs Selection: 12](#_Toc35848167)

[PowerShell Script Section: 13](#_Toc35848168)

[STIG / Vulnerability Properties: 13](#_Toc35848169)

[Finding Details / Comments: 14](#_Toc35848170)

[File / Buttons Section: 15](#_Toc35848171)

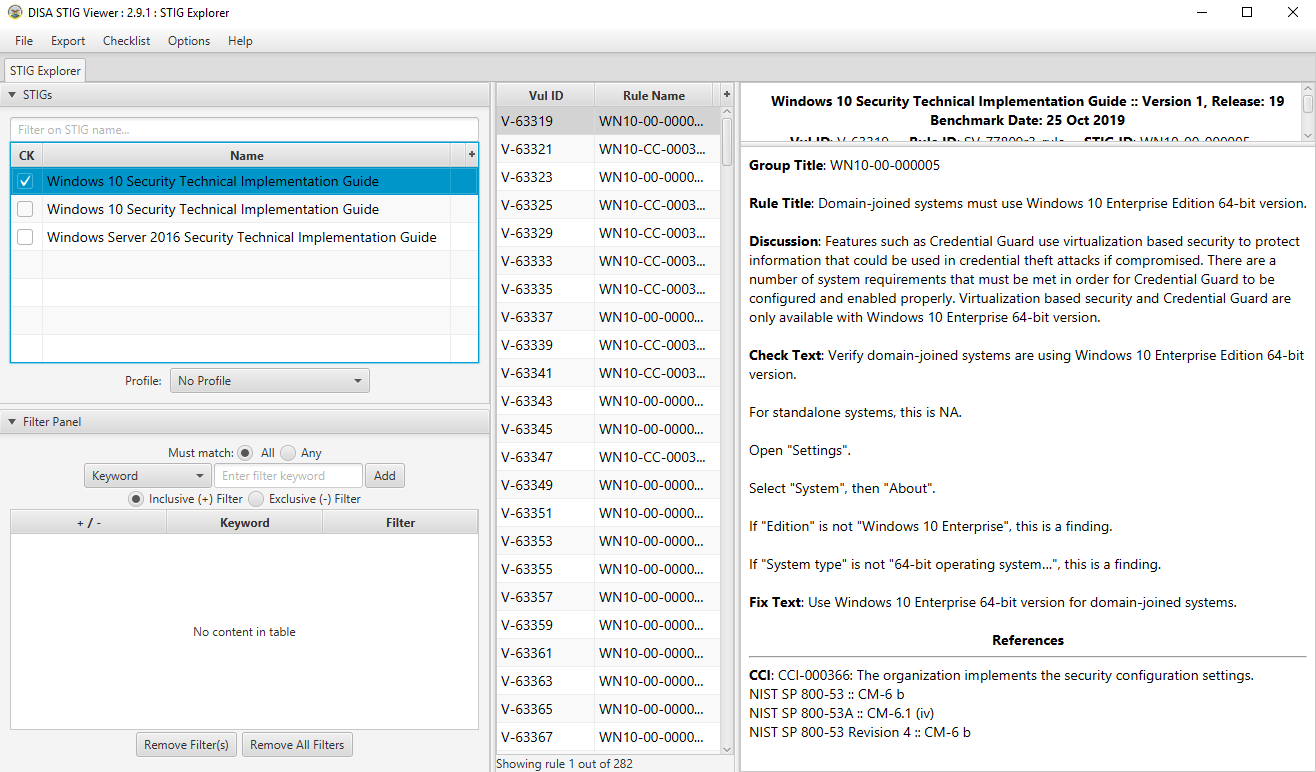
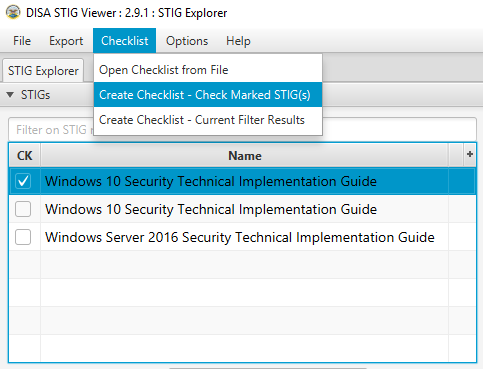
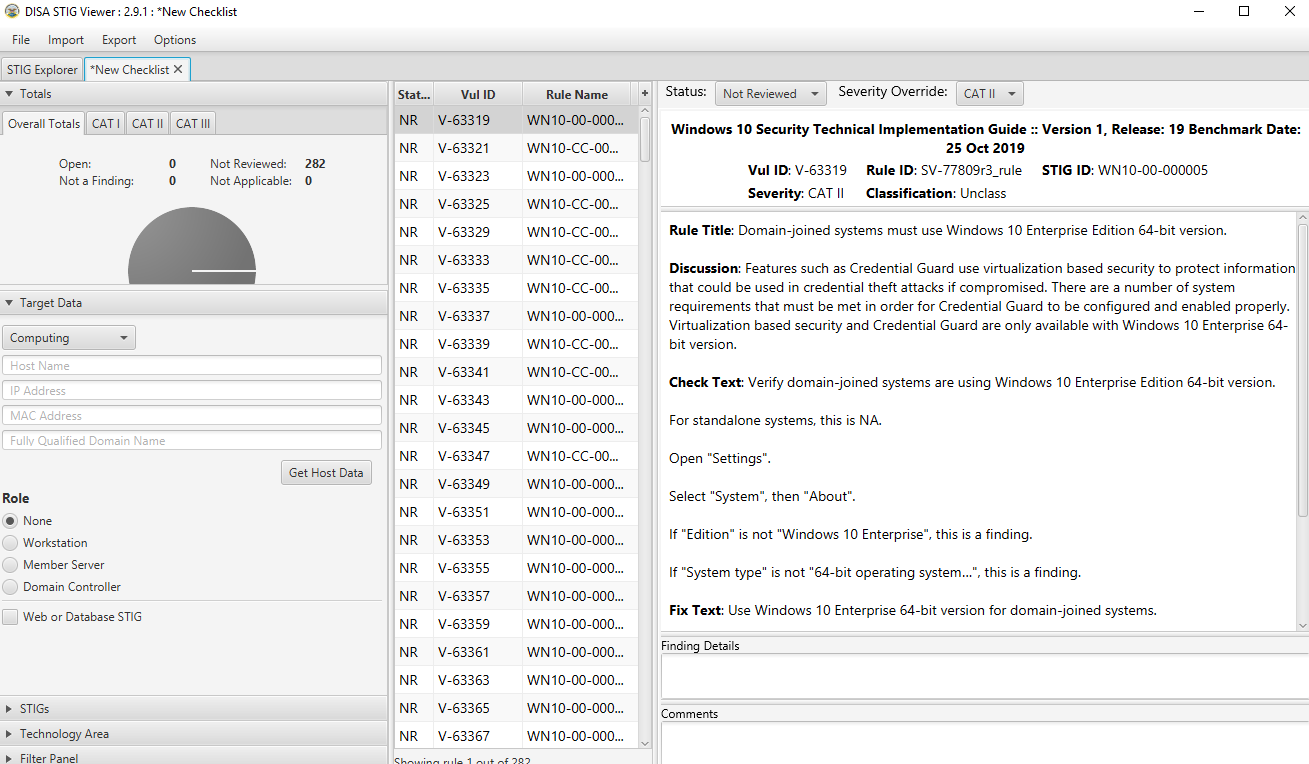
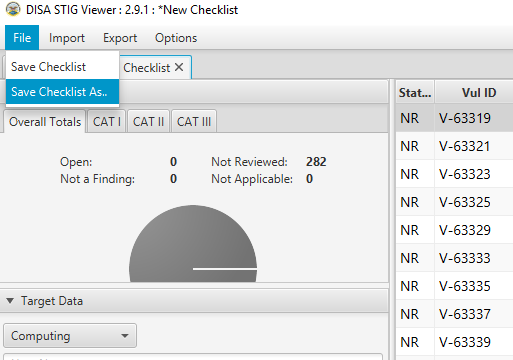
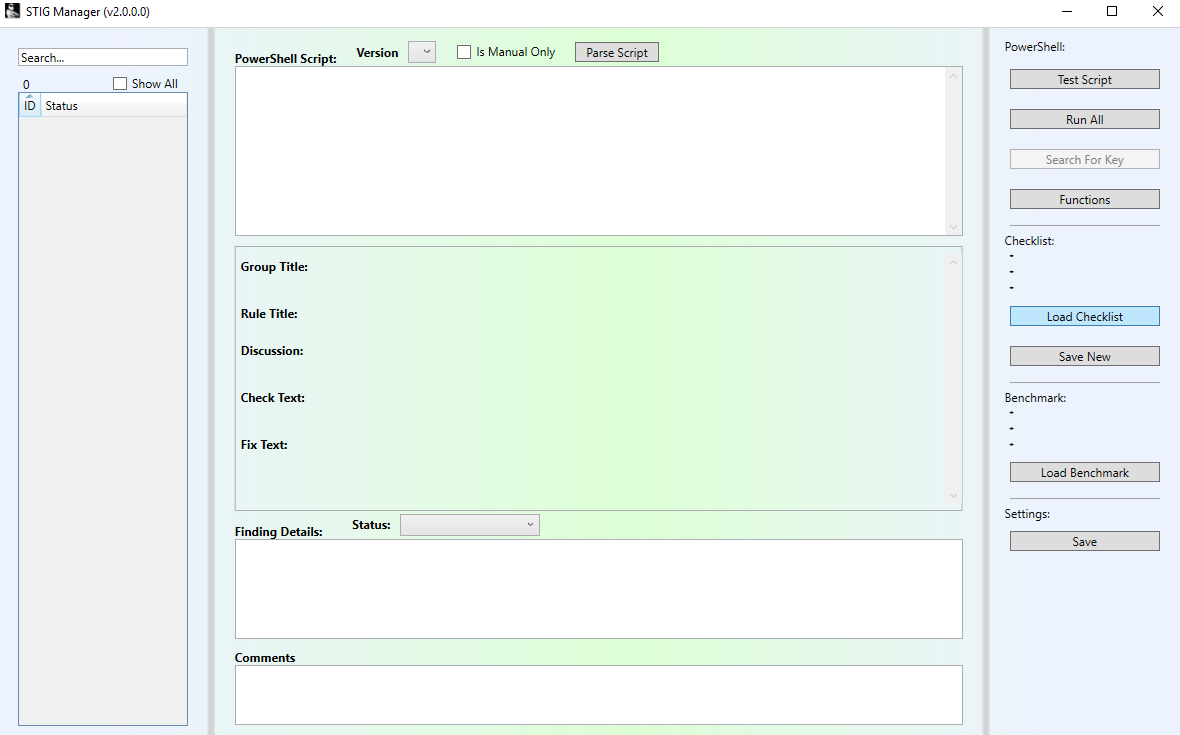
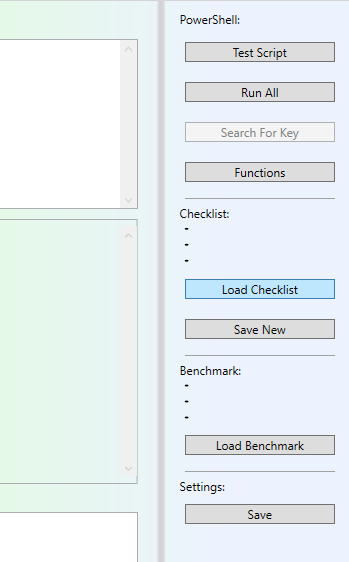
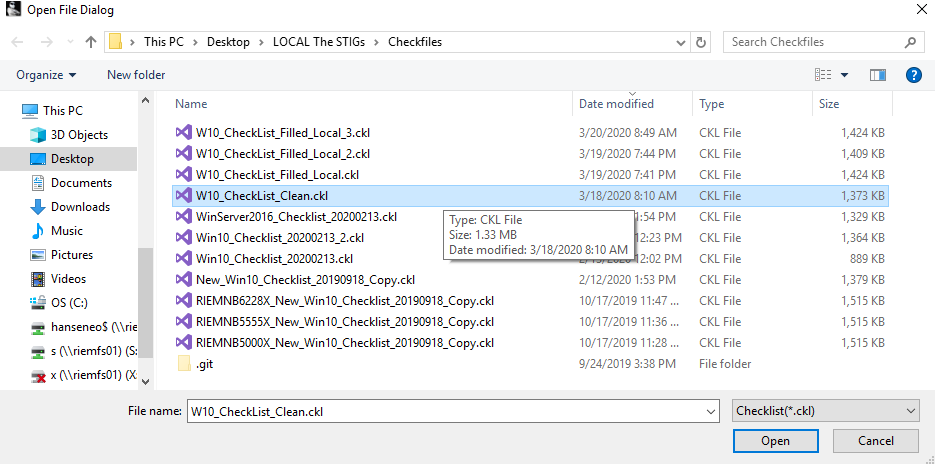
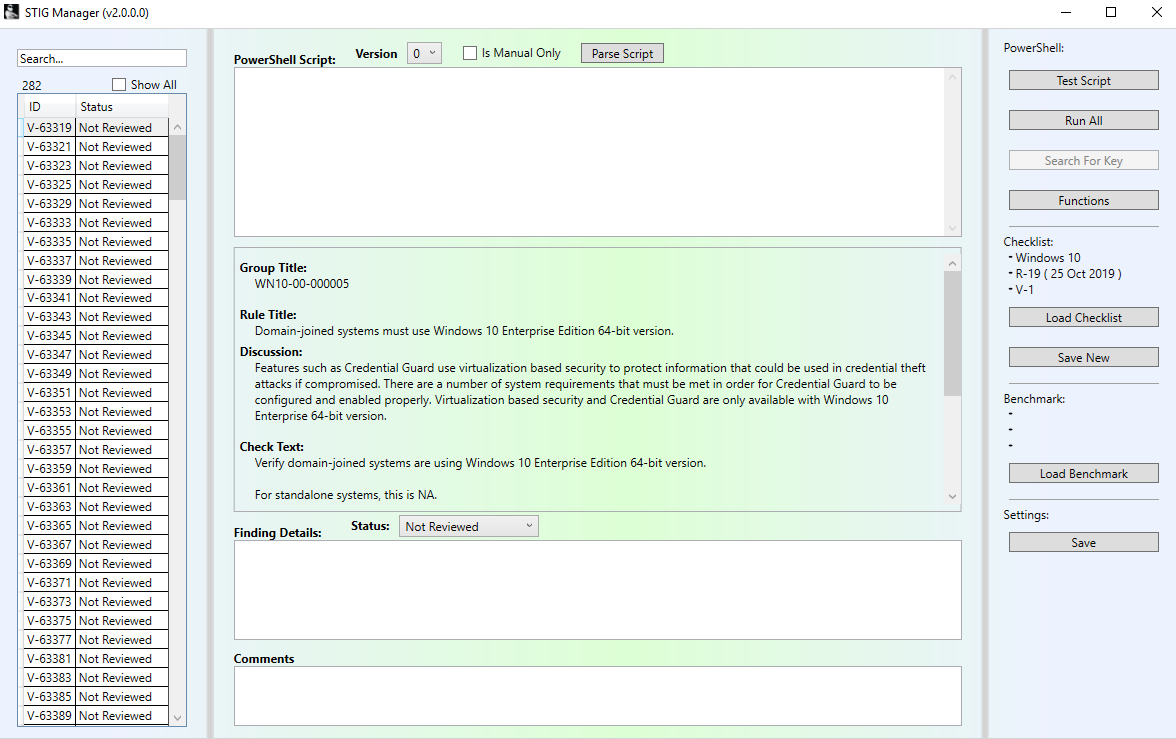
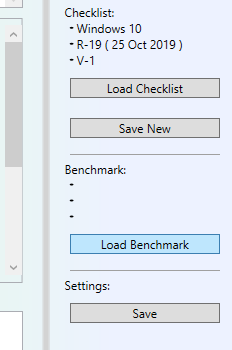
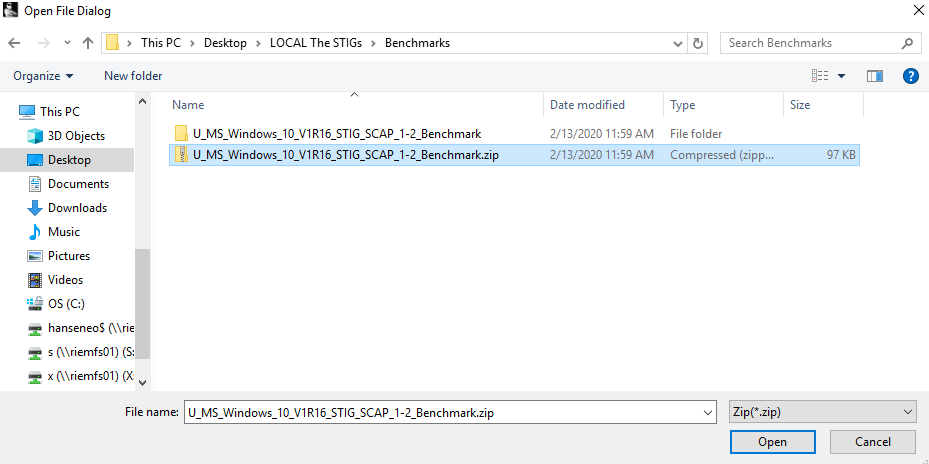
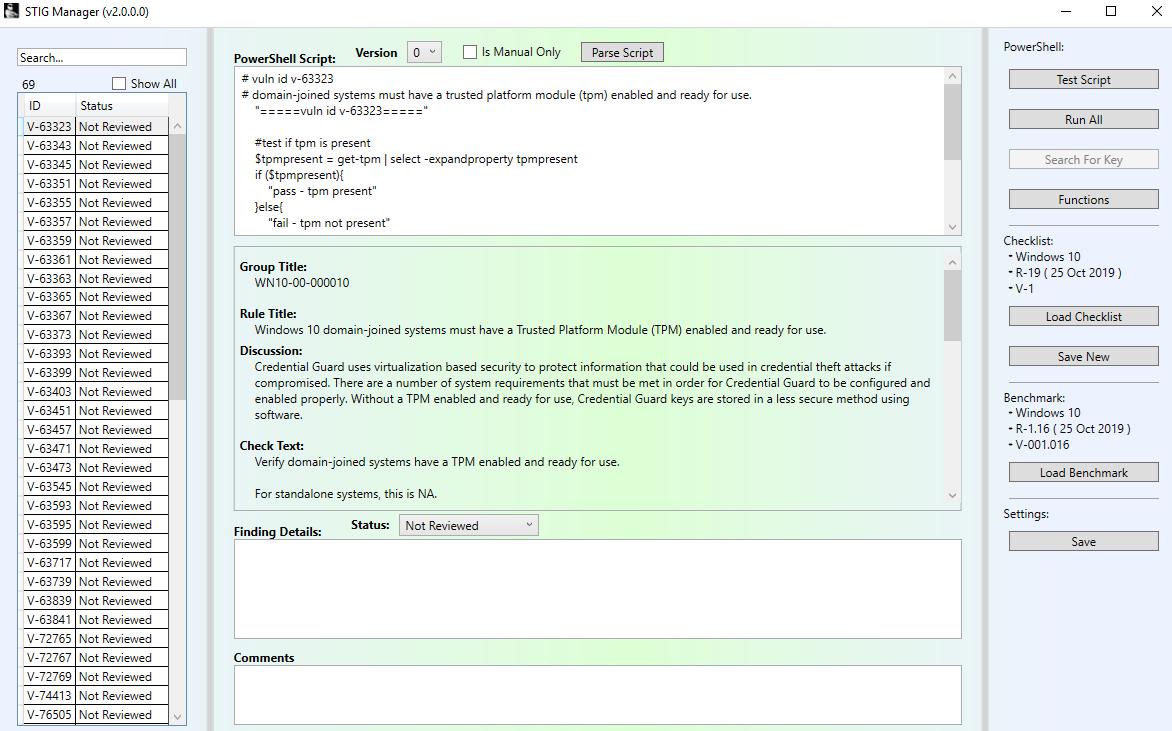
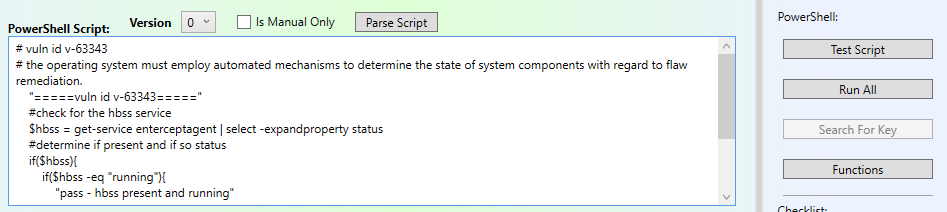
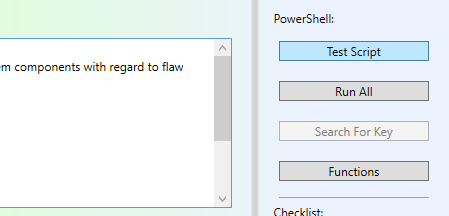
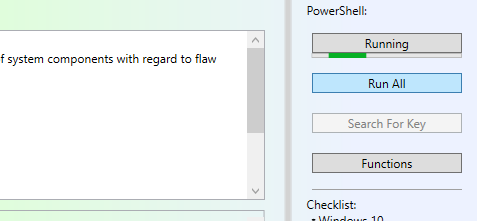
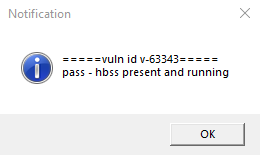
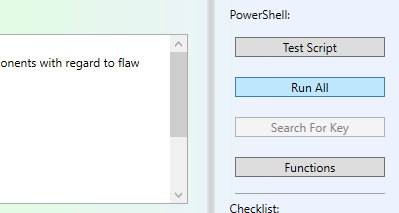
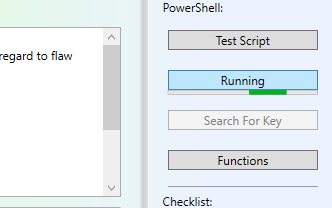
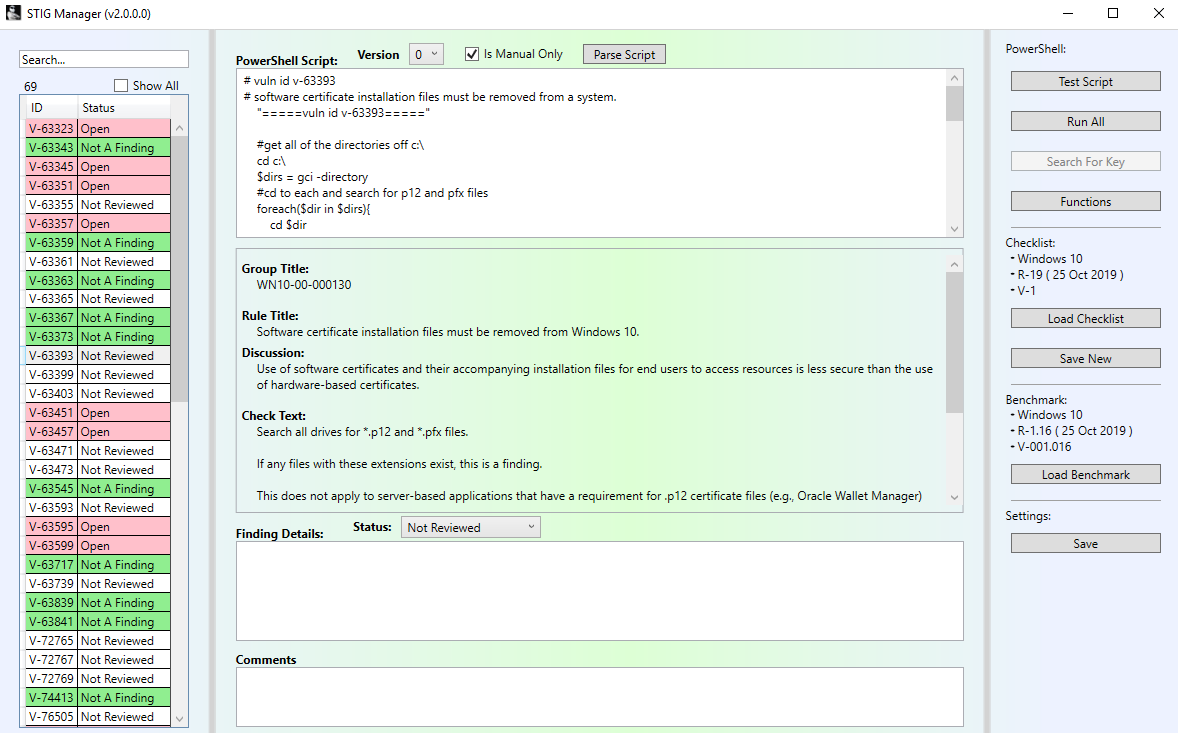
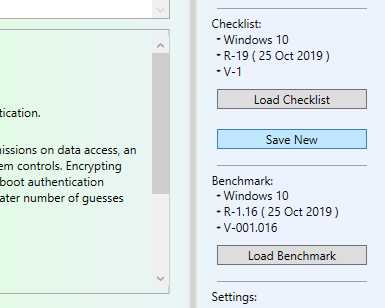
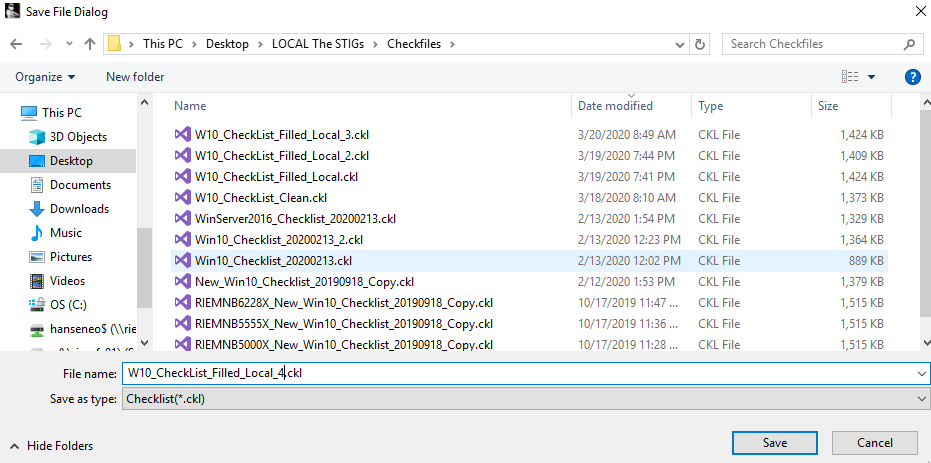
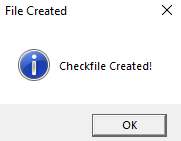
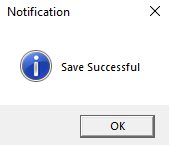
# About:

Welcome to the STIG Manager 2 reference documentation. The SM2(STIG Manager 2) gives users a way to import a checklist file from the STIG Viewer and build or use previously created PowerShell scripts to determine statuses for devices that the PowerShell scripts are ran on. Once the PowerShell scripts are ran on that device the SM2 stores the findings and allows the user to make comments and adjust the finding details for each STIG before running the given benchmark that corresponds to the current version of STIGs. Once the benchmark has finished running, the Checklist File should be fully populated with all findings.

SM2 has been developed by Eric Hansen. SM2 is a C# WPF application

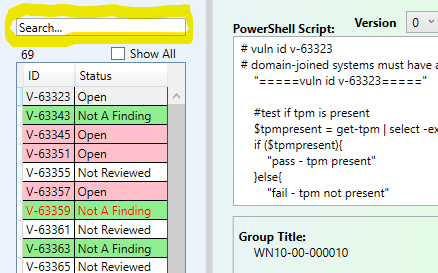
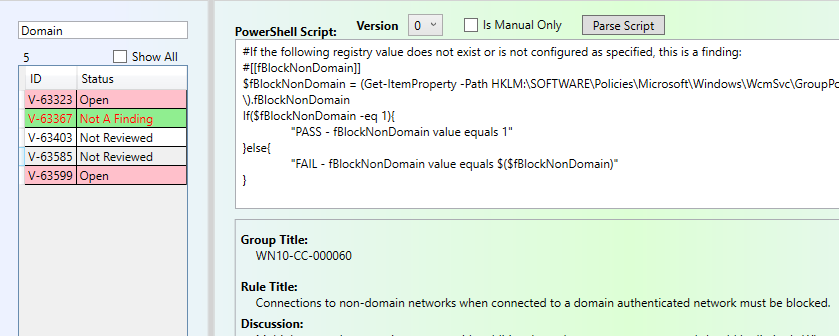
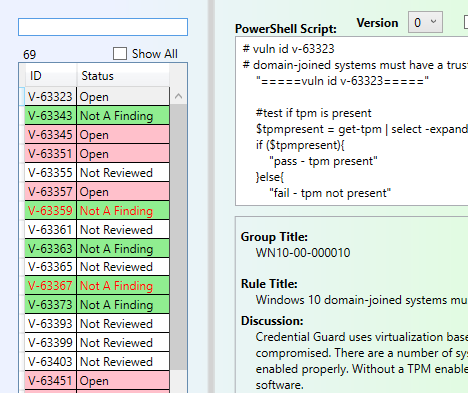
NOTE: These screenshots maybe different as the SM2 is being developed still.

# Basic Steps:

1. Open the STIG Viewer
   1. 
2. Open a STIG to view
   1. 
3. Create Checklist File and Save it.
   1. 
   2. 
   3. 
4. Open SM2
   1. 
   2. 
5. Import Checklist File into SM2
   1. 
   2. 
   3. 
6. Import Benchmark File into SM2
   1. 
   2. 
   3. 
7. Once the Checklist File and Benchmark File have been imported, begin creating PowerShell scripts to do STIG checks.
   1. 
8. Test the PowerShell script
   1. 
   2. 
   3. Displays Results:
      1. 
9. After testing PowerShell scripts, you can run all scripts against your local machine by clicking the “Run All” button.
   1. 
   2. 
   3. Results populate in the STIGs (Left column)
10. Save the Results into Checklist File
    1. 
    2. 
    3. 
    4. 
11. The SM2 is now displaying the Checklist File that you saved the results from, leaving the original checklist file how it was.
12. The Checklist File you created is now ready to have the benchmark ran on it.

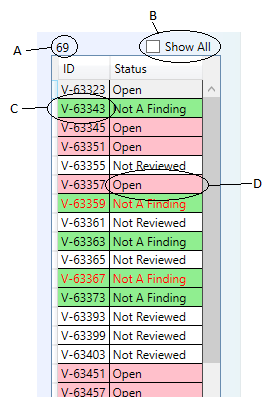
# Search:

The search box will look through the STIG ID, Status, Rule Title, and PowerShell Script. To use the search feature follow these steps:

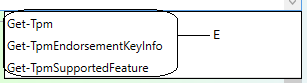
1. Click into the search textbox:
   1. 
2. Type what you are looking for:
   1. For Example I used the keyword “Domain”
   2. This will return all STIGs that has the keyword in the ID, Status, Rule Title, or PowerShell Script properties of that STIG.
   3. 
3. To clear your search, just delete the text from the search textbox.
   1. 

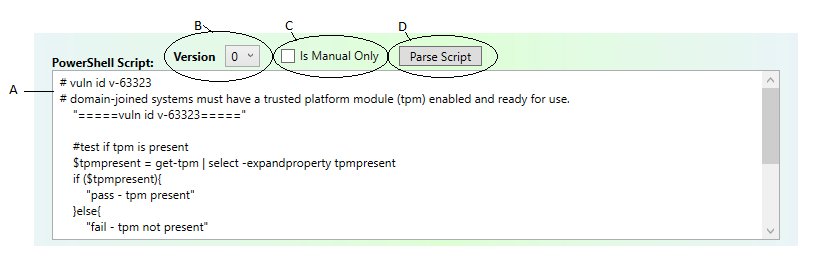
# STIGs Selection:

The STIGs selection is the left column that contains all the different vulnerabilities. In this column you will see each STIG represented by its ID and Status. Selecting one of these items will update the rest of the screen with that items current properties.

1. Number of STIGs (Vulnerabilities) being displayed.
2. Show All toggle, this will display all vulnerabilities in this Checklist File
3. Vulnerability ID
4. Vulnerability Status

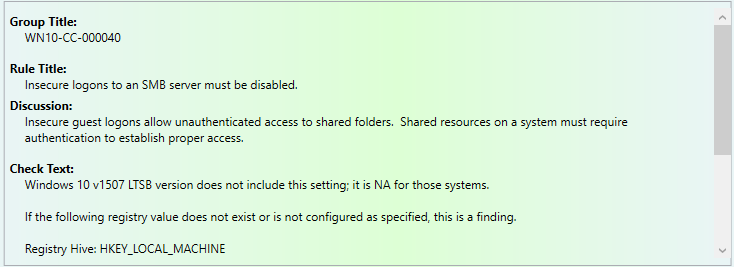
# PowerShell Script Section:

 The PowerShell Script Section is the main piece of the SM2. This section allows the user to build and edit PowerShell scripts to do checks on settings.

1. This section contains the actual PowerShell script that you want to run against the selected vulnerability / STIG
2. This is a drop-down selection that allows you to view previous version of the script you have been working on. Every time you make changes to the script and run / leave the textbox a new version is created.
3. “Is Manual Only” checkbox is for those PowerShell scripts that tend to take a long time to run. Checking this will cause this script **NOT** to run when the “Run All” button is pressed. This allows for the rest of the results to populate much quicker, because you don’t have to wait on particular scripts that take a long time.
4. The “Parse Script” button will look through the Check Content and try to parse any PowerShell scripts / Registry keys to check for and create a new version of the current PowerShell script.
5. This popup will display the PowerShell commandlet that you are trying to type out. To use this feature you can either click on the commandlet you want, or if it is the first entry in the box use the “Tab” key to complete the commandlet you are trying to write.

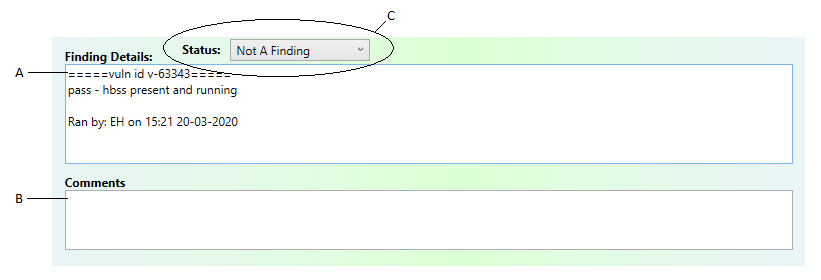
# STIG / Vulnerability Properties:

The STIG / Vulnerability Properties section contains all of the information needed to perform the STIG check / fix. This section contains the Group Title, Rule Title, Discussion, Check Text, and Fix Text.



# Finding Details / Comments:

The Finding Details / Comments section is the section in which the PowerShell script results are populated and you can create comments that will persist throughout all Checklist Files that you import. This section also allows you to change the Status of the currently selected STIG.

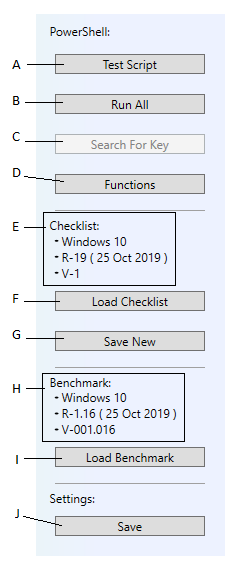


A. This is the Finding Details textbox, this textbox is populated by either the “Test Script” button or the “Run All” button.

B. This is the Comments textbox, these comments will persist throughout all Checklist Files imported.

C. The Status of the currently selected STIG / vulnerability, this drop-down allows you to manually change the status of this item.

# File / Buttons Section:

 The File / Buttons Section contains the major functionality of the SM2. This section allows the user to test PowerShell scripts, run all of the PowerShell scripts on the current computer, search the registry for a particular Registry value name path, add, remove, and edit PowerShell functions, load and save Checklist Files and Benchmark Files.

1. Test Script – Runs the current PowerShell script on the local computer.
2. Run All – Runs all of the current PowerShell scripts for each STIG on the local computer
3. The “Parse Script” button will look through the current script and search the Registry to see if the Value Name for a Registry path exists and add the paths to the script.
   1. Value Name that the Parse Script button looks for looks like this:
      1. Example (Exclude the quotes when adding to script): ”#[[RegistryValueName]]”
4. Functions – Opens a window that allows you to Add/Remove/Edit the PowerShell Functions that are used more than just once.
5. Checklist Info Panel – this panel displays the current Checklist File information
6. Load Checklist – allows you to load a different Checklist File
7. Save New / Save As – allows you to copy the original Checklist File and writes the current changes / results to the new file that is created and then loads that Checklist File as the current one.
8. Benchmark Info Panel – this panel displays the current Benchmark File
9. Load Benchmark – allows you to load a different Benchmark File
10. Save Settings – this will save the current state of the STIGs to the Database, the functions that were changed, the last Checklist File imported, and the last Benchmark File imported so that next time you open the SM2 it will resume right where you left off.